

```

100
    <schema xmlns="http://www.w3.org/2001/XMLSchema"
            xmlns:p="http://www.foo.com/personnel">
102
104
106
108
110
112
114
116
118
120
122
124
126
128
130
    <annotation>
    <documentation xml:lang="en">
        Personnel Directory Records for foo.com
        Copyright 2002 foo.com. All rights reserved.
    </documentation>
    </annotation>

    <element name='personnel'>
    <complexType>
    <sequence>
    <element name='employee' minOccurs='1' maxOccurs='unbounded'>
    <complexType>
    <sequence>
    <element name='lastname' type='string' maxOccurs='1'>
    <element name='firstname' type='string' maxOccurs='1'>
    <element ref='notes' minOccurs='0'>
    </sequence>
    <attribute name='serno' type='integer'>
    <attribute name='userid' type='p:USERID_TYPE'>
    <attribute name='department' type='string'>
    </complexType>
    </element>
    </sequence>

    <unique name='dummy1'>
    <selector xpath='personnel/employee'>
    <field xpath='@serno'>
    </unique>
    </complexType>
    </element>

    <element name="notes" type="string"/>

    <!-- Customized Simple data type declaration -->
    <simpleType name='USERID_TYPE'>
    <restriction base='string'>
    <pattern value="[a-zA-Z]{1}[1-9a-zA-Z]*"/>
    </restriction>
    </simpleType>
    </schema>
  
```

Diagram illustrating an XML Schema (XSD) structure with annotations pointing to specific elements and attributes:

- 100**: Points to the root `<schema>` element.
- 102**: Points to the `xmlns:p` namespace declaration.
- 104**: Points to the `<annotation>` element.
- 106**: Points to the `<documentation>` element.
- 108**: Points to the `<documentation>` element.
- 110**: Points to the `<documentation>` element.
- 112**: Points to the `<documentation>` element.
- 114**: Points to the `<documentation>` element.
- 116**: Points to the `<documentation>` element.
- 118**: Points to the `<documentation>` element.
- 120**: Points to the `<documentation>` element.
- 122**: Points to the `<documentation>` element.
- 124**: Points to the `<documentation>` element.
- 126**: Points to the `<documentation>` element.
- 128**: Points to the `<documentation>` element.
- 130**: Points to the `<documentation>` element.

FIGURE 1

```
<personnel>:= PS <employee> PE  
<employee>:= <employee>  
<employee>:= <employees> <employee>  
<employee>:= ES <lastname> <firstname> <notes_01> EE  
<lastname> := LS LE  
<firstname> := FS FE  
<notes>      := NS NE  
<notes_01> := epsilon | <notes>
```

FIGURE 2

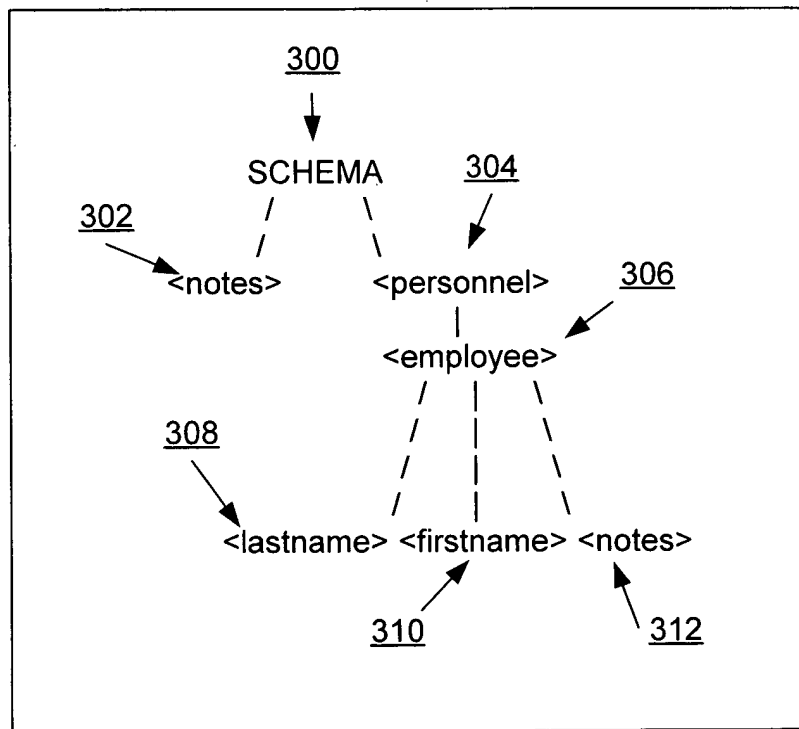


FIGURE 3

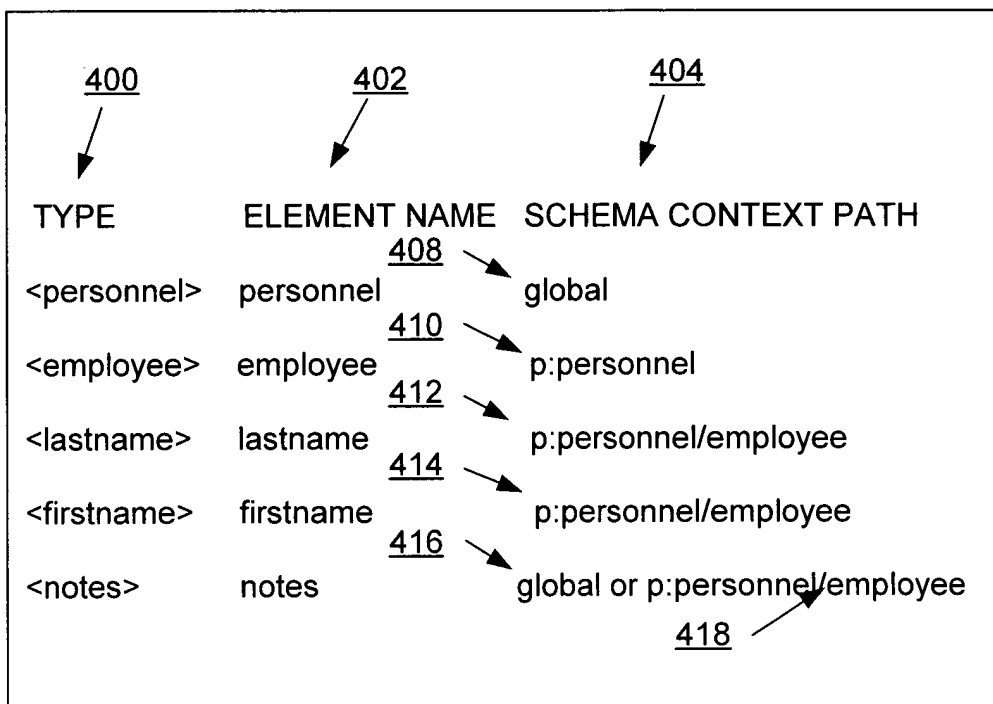


FIGURE 4

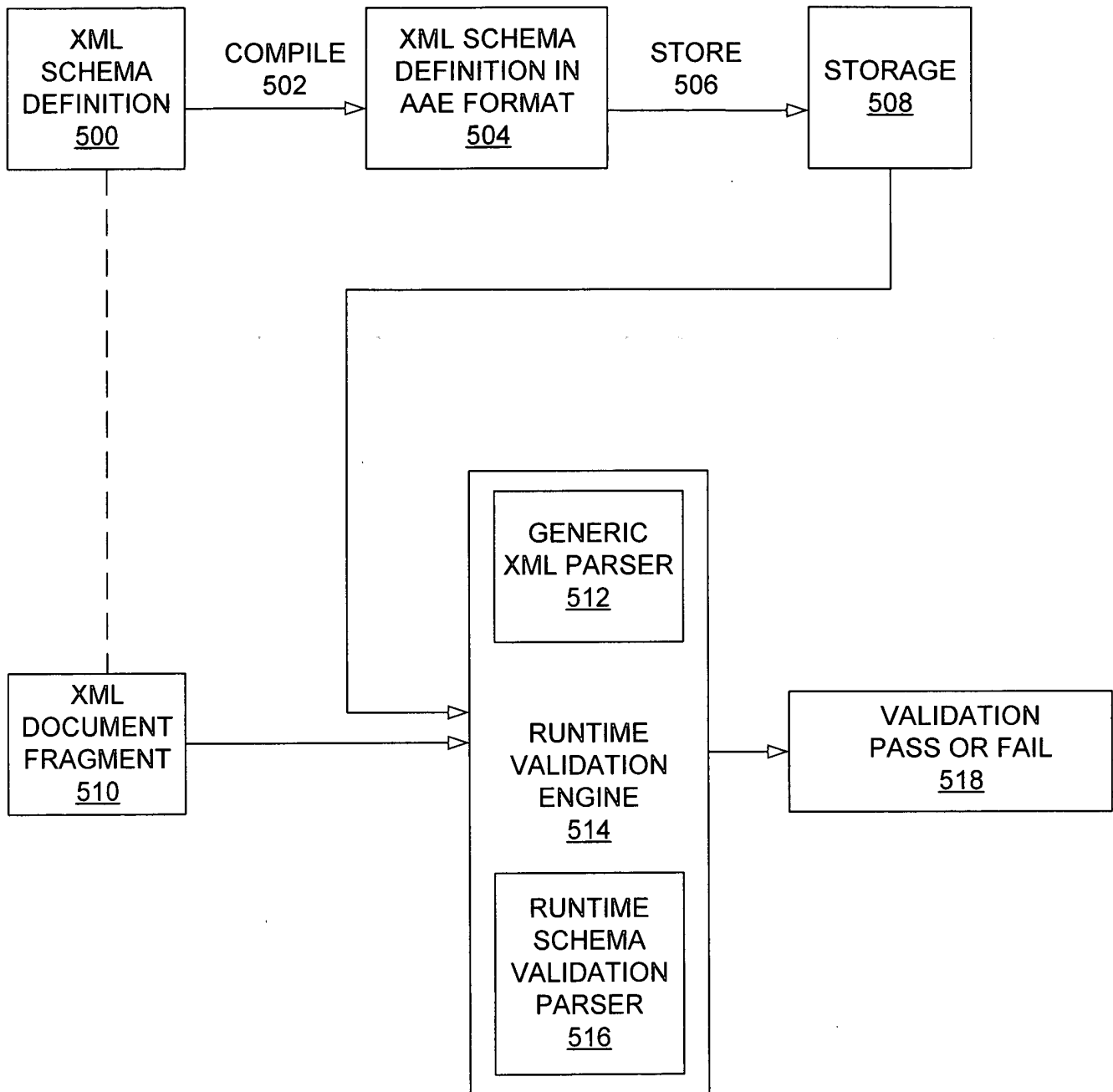


FIGURE 5

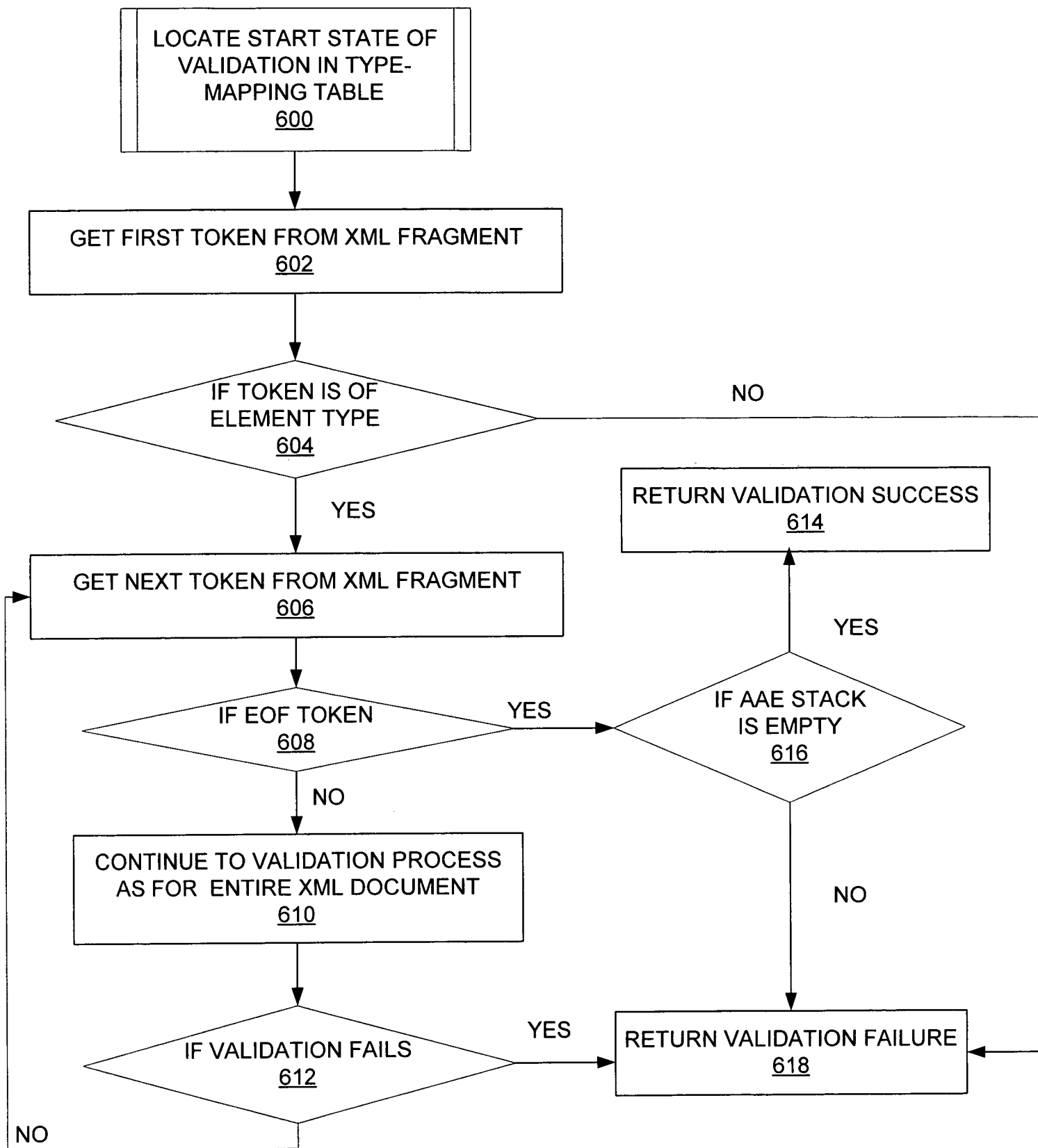


FIGURE 6

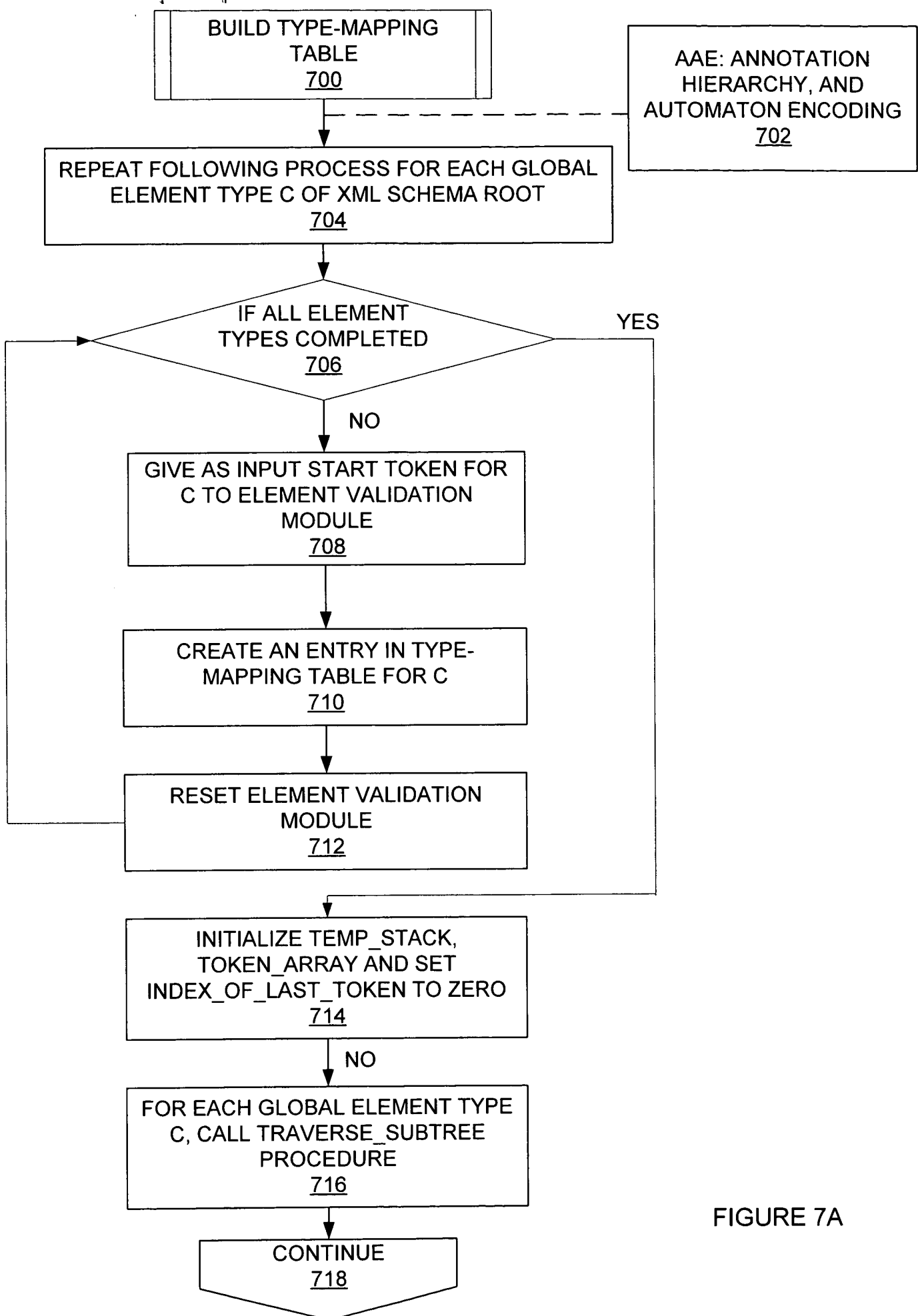


FIGURE 7A

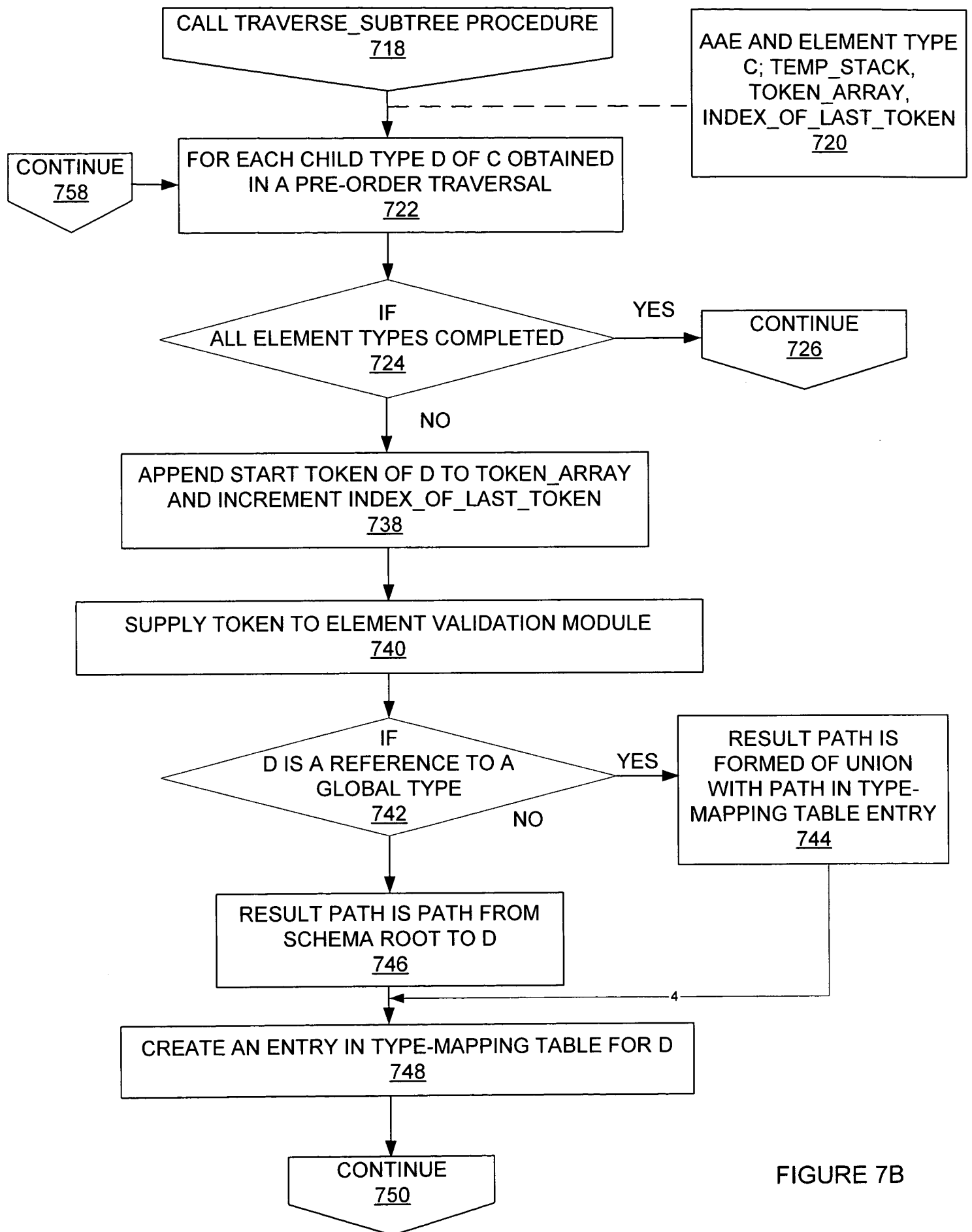


FIGURE 7B

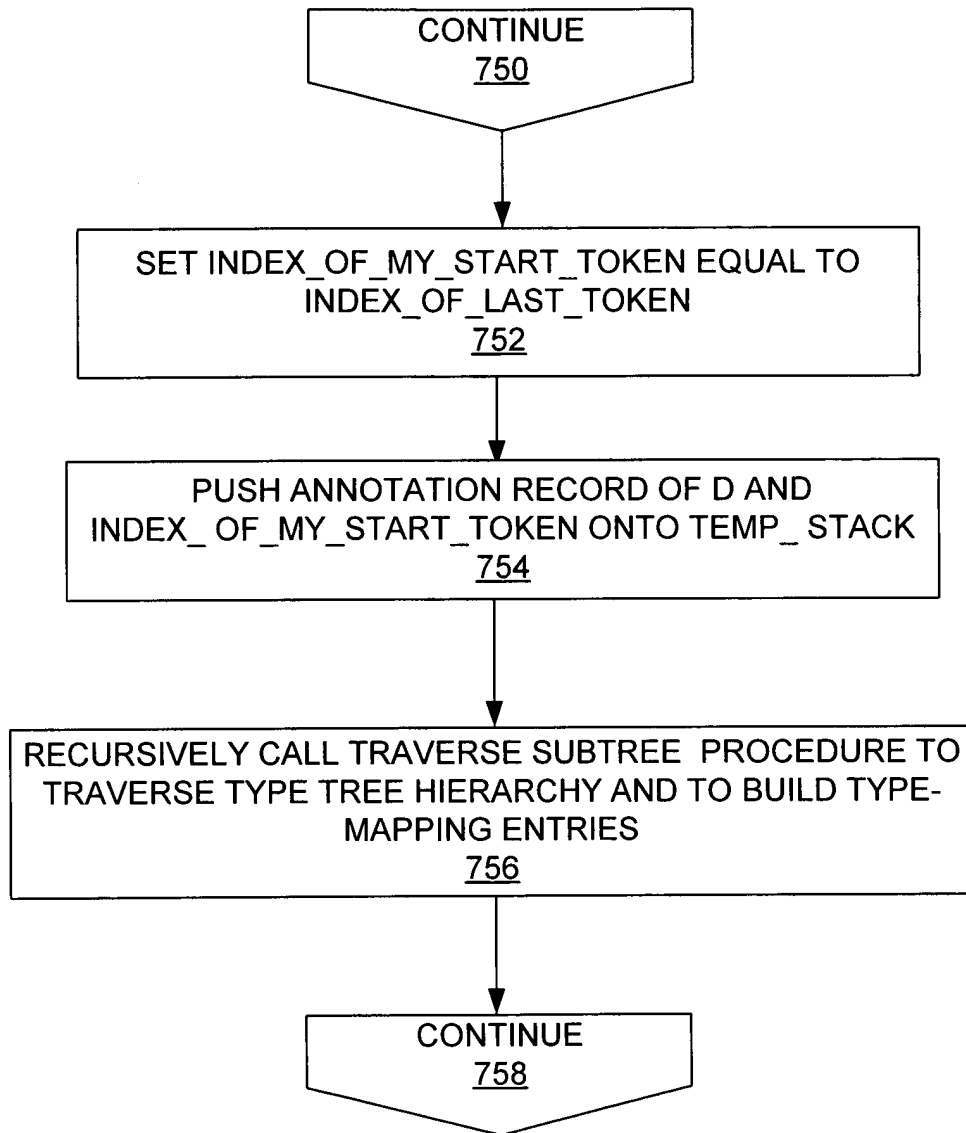


FIGURE 7C

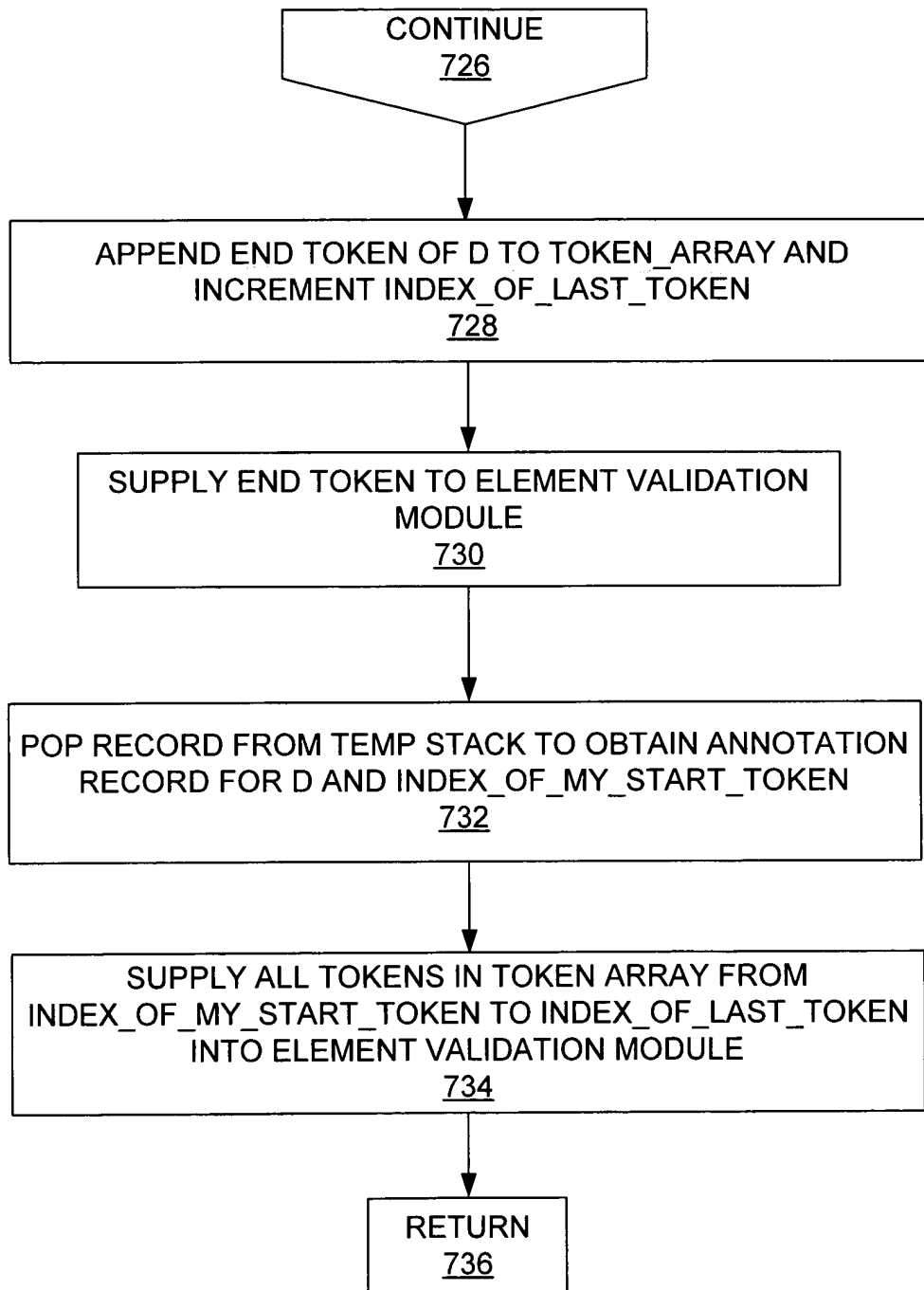


FIGURE 7D

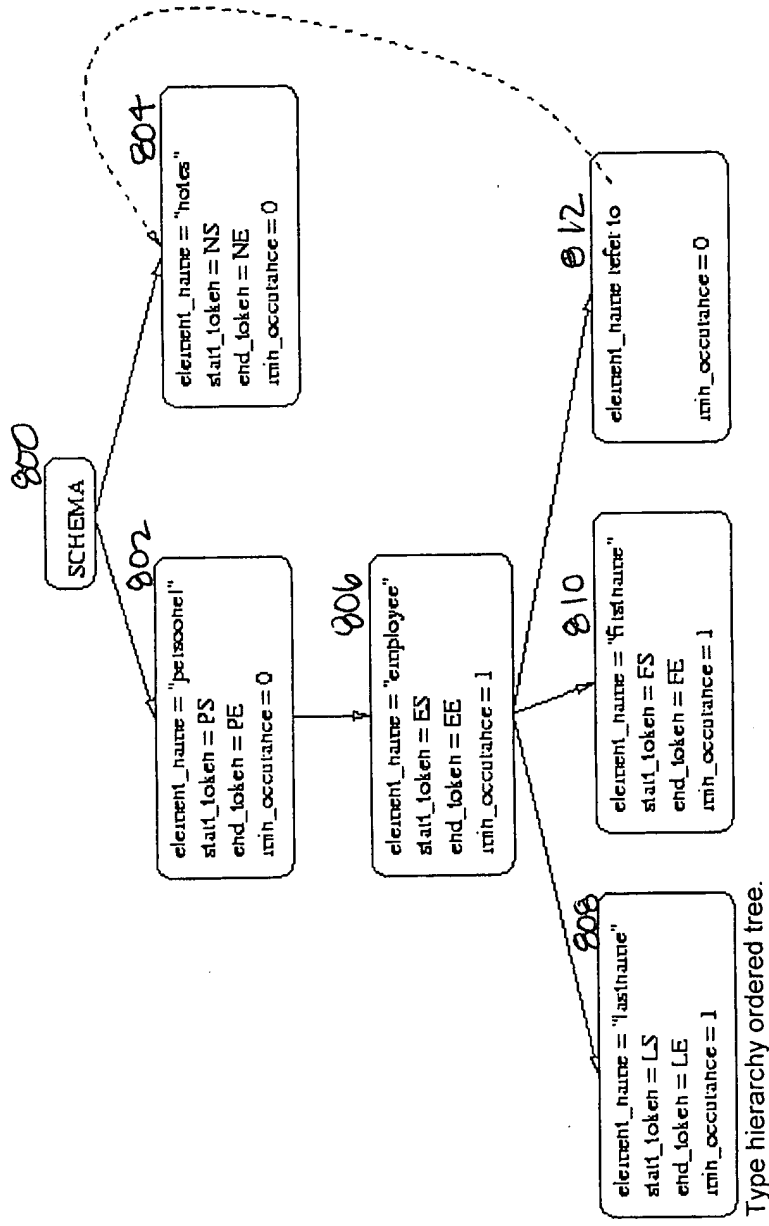


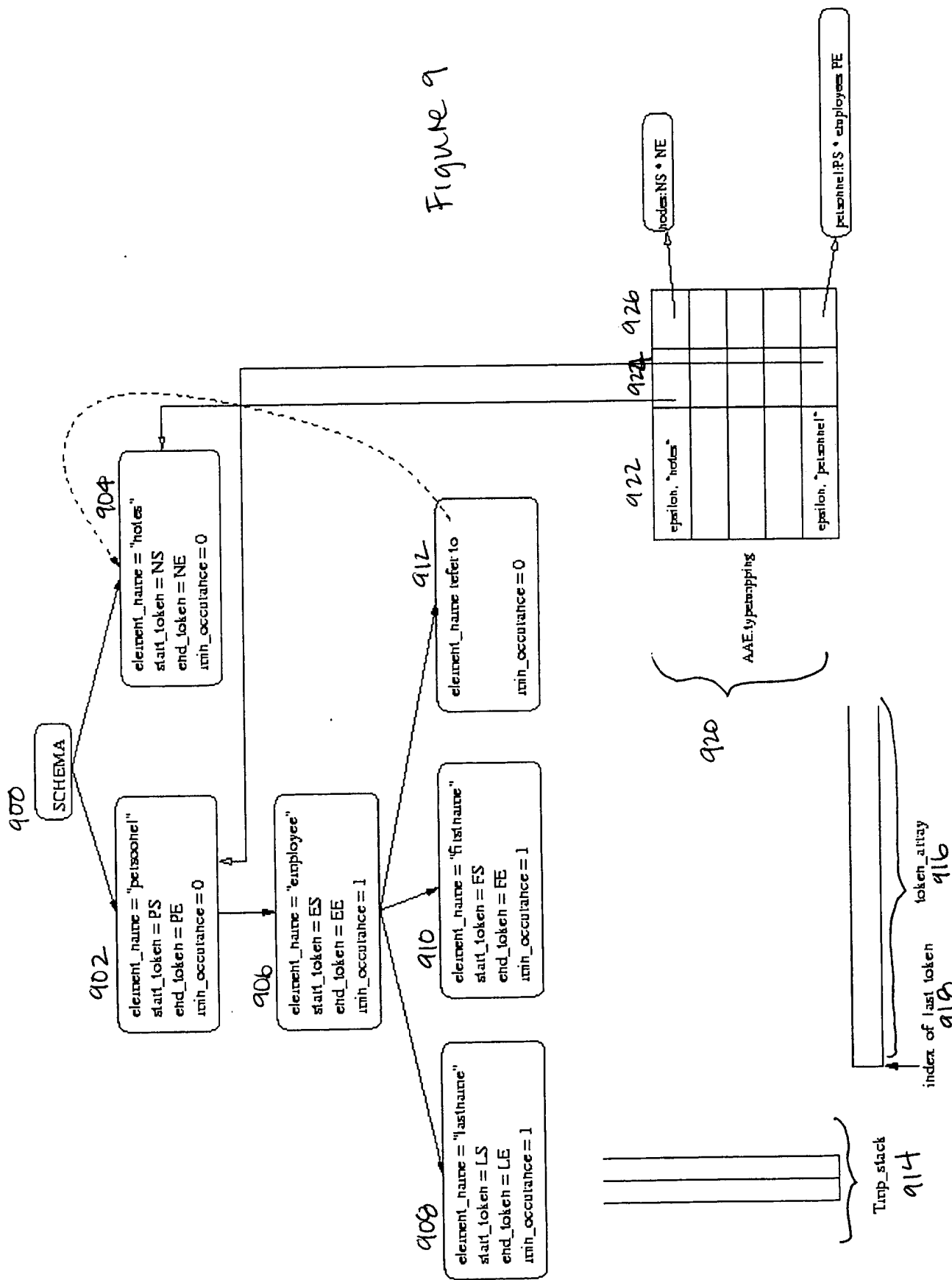
Figure 8

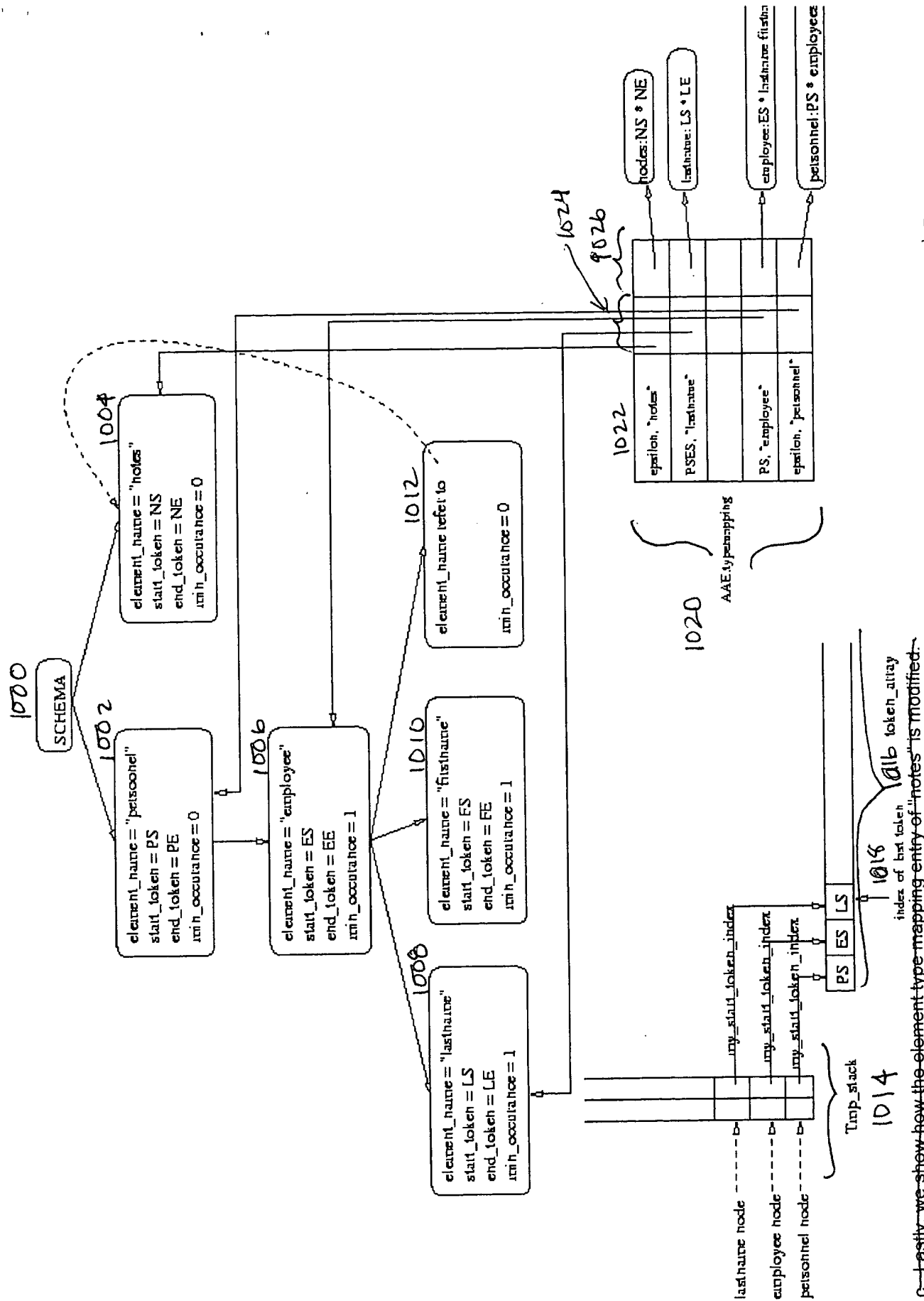
2. The following algorithm will traverse the above type hierarchy ordered tree once and create the type mapping table.

TRAVERAL_TYPE_HIERARCHY_TREE (tree node: C, Array of Token: Token_array, Int: index_of_last_token)

```

{
  for (each child node D of C in schema order)
  {
    APPEND start_token of D to the Token_array.
    INCREASE index_of_last_token such that it points to the appended token in token_array.
    FEED the above token into the PushdownAutomaton.
    if (D is a reference to a global element type)
    {
      /* C */
      Union the path from SCHEMA to D with the path in the type mapping entry of the global element type.
      replace the original path with the result path above.
    }else
    {
      create an entry <the path from SCHEMA to D, the element name of D, type annotation of D, the current state of the PushdownAutomaton>.
      my_start_token_index := index_of_last_token.
      PUSH the annotation of D and my_start_token_index onto Tmp_stack.
      TRAVERAL_TYPE_HIERARCHY_TREE (D, Token_array, index_of_last_token);
    }
  }
}
  
```





c. ~~Lastly, we show how the element type mapping entry of "notes" is modified.~~

1100

SCHEMA

1102

element_name = "personnel"
start_token = PS
end_token = PE
min_occurrence = 0

1104

element_name = "notes"
start_token = NS
end_token = NE
min_occurrence = 0

1106

element_name = "employee"
start_token = ES
end_token = EE
min_occurrence = 1

1108

element_name = "lastname"
start_token = LS
end_token = LE
min_occurrence = 1

1110

element_name = "firstname"
start_token = FS
end_token = FE
min_occurrence = 1

1112

element_name refer to
min_occurrence = 0

1124

(PSES)*, "notes"						nodes: NS * NE
PSES, "lastname"						lastname: LS * LE
PSES, "firstname"						firstname: FS * FE
PS, "employee"						employee: ES * EE
personnel, "personnel"						personnel: PS * PE

1120

AAE type mapping

my_start_token_index

my_start_token_index

my_start_token_index

PS	ES	LS	LE	FS	FE	NS
----	----	----	----	----	----	----

token array

1118

index of last token

1116

Temp stack

1114

notes node

employee node

personnel node

Figure 11